International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)
ISSN (P): 2249-6890; ISSN (E): 2249-8001
Vol. 7, Issue 2, Apr 2017, 155-166

A HYBRID ALGORITHM USING FIREFLY AND CUCKOO SEARCH ALGORITHM FOR FLEXIBLE OPEN SHOP SCHEDULING PROBLEM

N. KAMATCHI[1] & D. SARAVANAN[2]

[1]*Research Scholar, Bharathiyar University, Coimbatore, Tamil Nadu, India*

[2]*Professor of Mathematics, Karpaga Vinayaga College of Engineering and Technology, Chennai, Tamil Nadu, India*

ABSTRACT

In this paper presents the hybrid algorithm using firefly and a cuckoo search algorithm for flexible open shop scheduling problem. The flexible, open shop scheduling is known to be NP-hard. Cuckoo algorithm (CA) is one of the widely used techniques for constrained optimization. And it gave the best results compared to other algorithms. A disadvantage of cuckoo algorithm though is that they easily become trapped in the local minima. In this paper, a firefly algorithm (FA) is used with a cuckoo algorithm in order to avoid the local minima problem and to improve the best results. The proposed combined hybrid algorithm was compared with ACO (Ant Colony Optimization), GA (Genetic Algorithm) to prove the importance of the hybrid algorithm. It minimizes the make span time and the scheduling can be used in scientific computing.

KEYWORDS: Flexible Open Shop Scheduling, Firefly Algorithm, Cuckoo Algorithm, Makes Span and Minimization

Received: Feb 01, 2017; **Accepted:** Mar 17, 2017; **Published:** Mar 21, 2017; **Paper Id.:** IJMPERDAPR201715

Original Article

I. INTRODUCTION

Open shop scheduling problem (OSSP) is the hardest combinatorial optimization problems in the branch of production scheduling. It is an important genre of NP-hard problems and requires extensive computational resources of schedules a set of jobs on a set of machines with the objective to minimize certain, subjected to the constraint that each job has a specified processing order through all machines which are fixed and known in advance. Flexible open shop scheduling problem (FOSSP) is an extension of the traditional OSSP that allows one operation which can be processed one machine out of a set of alternative machines. This type of flexibility in scheduling problem increase searching scope of real life solutions. It is applicable for flexible manufacturing situations. In this section n jobs and m

Machines of FOSSP have been described as to minimize make span (i.e.,) the completion time of processing all jobs by scheduling n by m (3). In this paper, we have proposed a hybrid algorithm which combines the advantages of cuckoo search algorithm and firefly algorithm so as to solve flexible open shop scheduling problems. FOSSP can be used in scientific computing and high power computing for solving all the optimization problems (5).The performance comparison of the hybrid firefly algorithm and cuckoo algorithm is compared by make span time.

The important contribution of this paper is as follows

- The proposal of a Hybrid algorithm which combines the advantage of CA and FA.

- The performance comparison of the Hybrid algorithm with genetic algorithm and ant colony optimization.

II. LITERATURE SURVEY

The firefly algorithm (FA) is a novel Meta heuristic algorithm inspired by the social behavior of fireflies (10). The cuckoo algorithm (CA) has been proven to deliver excellent performance in function optimization, neural network training and engineering design. Combining the CA and FA an approach for computational intelligence technique for solving FOSSP is framed. (3) Authors (Ala'a Abu-srhahn et al,2015) presented a hybrid algorithm using cuckoo algorithm and genetic algorithm for job shop scheduling problem(3).He proposed a hybrid algorithm to minimize the make span for JSSP. Mahashwar et al (9) 2014 presented firefly algorithm based improved genetic algorithm. He also proposed an approach where the initial population is selected from a pool of population on the basis of fire fly algorithm. Shaik Farook, P.saangemeswara Raju et al(6) investigated a evolutionary hybrid genetic-firefly algorithm for global optimization. He was implemented this on various bench marking test functions to evaluate the general performances of the individual algorithms and hybrid genetic- firefly algorithm interms of rate of convergence.

Ravianandan. M et al,2015 (2) presented a hybrid flow shop scheduling using improved ACO and CA to minimize make span doing so, They observed a better results than all the problems in terms of the minimum make span. Satyendra Singh et al, 2015 (5) proposed a hybrid genetic and cuckoo search algorithm for job scheduling. He trapped easily local optima of disadvantage of genetic algorithm using cuckoo algorithm. Lina Zhang et al, 2016 (6) investigated a novel hybrid firefly algorithm for global optimization. They compared results with the original version of the firefly algorithm (FA), differential evolution (DE) and PSO in the sense of avoiding local minima and increasing the convergence rate. Hossein Parvan et al (2014) (10), explained a new hybrid algorithms for task-scheduling in computational grids to decrease makespan. He proposed a hybrid algorithm for solving the independent task scheduling problem in grid. In (S. karthikeyan et al, 2014) (10) a hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints has been proposed by the authors. They gave better results than other author's algorithm.

Xin-She Yang (12) developed and explained firefly algorithm and Xin-She –Yang and Deb developed the cuckoo search algorithm and provided an insight for solving minimization objective function with constrained parameters using these algorithms. Kanagaraj et al, (2013) presented a hybrid algorithm of genetic algorithm and cuckoo algorithm for reliability redundancy allocation problems. They observed a better balance between exploration and exploitation. R. G. Babukartik and P. Dhava chelvan explained a hybrid algorithm for solving job scheduling problem using advantage of both genetic and cuckoo algorithm. They showed the size of the problem increases task of the problem increases task creation time and result retrieval time also increases. Maryan rabiee et al used a cuckoo search algorithm for job scheduling in grid computing. In this paper author compared the result of cuckoo algorithm, genetic algorithm with PSO and show that CA can complete tasks, minimum time as compared to other GA and PSO.

III. PROPOSED WORK

- **Fire Fly Algorithm**

Fire fly algorithm (FA) [1,2] is one of the latest a meta heuristic as it is inspired by the flashing behaviour of fireflies. It is the most charismatic of all insects. The Fire fly Algorithm (FA) is a population –based technique to find the global optimal solution based on swarm intelligence, investigating the foraging behavior of fireflies. Fire flies, which

belong to the family of lampyridae, are tiny winged beetles having the capability of producing light with little or no heat and it is called a cold light. It flashes the light in order to attract mates. They are whispered to have a capacitor-like mechanism, that gradually charges until the definite threshold is reached, at which they discharge the energy in the form of light,subsequent to which the cycle repeats.

It was developed by Dr. Xin –She Yang at Cambridge University in 2008.It is based on the firefly bugs behavior, including the light emission, light absorption and the mutual attraction, which was developed to solve the continuous optimization problems. The flashing light of fireflies is a powerful sight in the summary sky in the tropical and temperate regions. There are more number of firefly spices and most fireflies produce short and rhythmic flash. The model of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence and the true functions of such signaling systems are still debating. However, two fundamental functions of such flashes are to attract mating partners and to attract potential prey. In addition, flashing may be serving as a protective warning mechanism.

The rhythmic flash, the rate of flashing and the amount of time from part of the signal system that brings both sexes together. Females respond to a male's unique pattern of flashing in the same spices, while in some species such as photuris, female fireflies can mimic the mating flashing pattern of other species so as the lure and eat the male fireflies who may mistake the flashes as a potential suitable mate. The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes; it's possible to formulate new optimization algorithms. When nature inspires algorithm such as particles and swarm optimization (PSO) [14,15] as firefly algorithm are the most power fly algorithm for optimization.

- **Structure of the Firefly Algorithm**

Compared to the other evolutionary algorithms, FA has many advantages in solving complex optimization problems. This algorithm is based on a physical formula of light intensity L that decreases with the increase in the square of the distance d^{2}.. However, as the distance from the light source increases, the light absorption causes that light becomes weaker and weaker. This can be associated with the objective function to be optimized.

The development of firefly-inspired algorithm was based on following idealized rules [1, 18]

- All fireflies are unisex so that sex is not an issue for attraction.
- Their attractiveness is proportional to their flashing brightness.
- The light intensity of a fire fly is affected and determined by the landscape of the fitness function.

The firefly S attracts all other fireflies and is attracted to all other fireflies, The less bright firefly is attracted and moved to the brighter one, The brightness decreases when the distance between fireflies is increased, The brightest firefly moves randomly (no other fireflies can attract it), The firefly particles are randomly distributed in the search space.

According to above rules there are two main points in firefly algorithm, the attractiveness of the firefly and the movement towards the attractive firefly [7]

Makespan Calculation

Objective Function is to minimize makesapn i.e to minimize time between start of first work and completion of last work

C_{max} = Max (C_{i-1j},C_{j-1i})+P_{ij}

Where

C_{i-1j},C_{j-1i} = Completion time of previous operation

P_{ij}= Processing time of next operation

Subject to machine availability in each stage to process the job.

- **Characteristics of the Firefly Algorithm**

In firefly algorithm, there are two important issues need to be defined: the first one is the variation of light intensity and the second one is the formulation of the attractiveness. We can assume always that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function [21].

In the simplest case for maximum optimization problems, the light intensity L of a firefly representing the solution S is proportional to the value of fitness function

L(S) α f(S)

However, the attractiveness β is relative; it should be seen in the eyes of the beholder or judged by the other fireflies. Thus, it will vary with the distance d_{ij} between firefly I and firefly j. In addition, light intensity (L) decreases with the distance from its source, and light is also absorbed in the media, so we should allow the attractiveness to vary with the degree of absorption. In the simplest form, the light intensity (L_d) varies according to the inverse square law L(d)= L_s/d^2 where L_s is the intensity at the source. For a given medium with a fixed light absorption coefficient ϒ, the light intensity L varies with the distance d. i.e. L = L_0 e $^{-ϒd}$, where L_0 is the original light intensity. In order to avoid the singularity at d=0 in the expression L_s/d^2, the combined effect of both the inverse square law and absorption can be approximated using the following Gaussian form

$$\mathbf{L(d) = L_0\, e^{-ϒ d^2}} \quad (2)$$

Sometimes, we may need a function which decreases monotonically at a slower rate. In this case, we can use the following approximation

$$L(d)= \frac{L_0}{(1+\gamma d^2)} \quad (3)$$

As a fire fly's attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by [8]

$$\beta\,(d) = \beta_0\, \mathbf{e}^{-ϒ d^2} \quad (5)$$

Where β_0 is the attractiveness at d =0. And ϒ is the light absorption coefficient which is fixed during the execution of the algorithm.

In the implementation, the actual form of attractiveness function β (d) can be any monotonically decreasing functions such as the following generalized form

β (d) = $\beta_0 e^{-\gamma d^{n}}$, (n ≥ 1)

For a fixed ϒ, the characteristic length becomes r = $ϒ^{-1/n}$ → 1 as n →∞.

Conversely, for a given length scale r in an optimization problem, the parameter ϒ can be used as a typical initial value that is $\Upsilon = 1 / r^{n.}$

The distance d between any two firefly i and j at positions S_i and S_j respectively is the Cartesian distance [8, 13]

$$d_{ij} = \| S_{i,h} - S_{j,h} \| = \sqrt{\sum_{h=1}^{d} (S_i - S_j)^2}$$

Yang (2010) described the movement of a firefly i at position S_i moving to a brighter firefly j at position S_j by

$$S_i(t+1) = S_i(t) + \beta_0 \mathbf{e}^{-\Upsilon d^2} (S_i(t) - S_j(t)) + \alpha \varepsilon_i \quad (6)$$

Where $\beta_0 \mathbf{e}^{-\Upsilon d^2} (S_i(t) - S_j(t))$ is due to the attraction of the firefly S_j and $\alpha \varepsilon_i$ a randomization parameter ; so if $\beta_0 = 0$ then it turns out to be a simple random movement [8,13].

The algorithm compares the attractiveness of the new firefly position [22] with old one.If the new position produces higher attractiveness value, the firefly is moved to the new position; otherwise the firefly will remain in the current position. The termination criterion of the FA is based on an arbitrary predefined number of iterations or predefined fitness value. The brightest firefly moves randomly based on the following equation

$$S_i(t+1) = S_i(t) + \alpha \varepsilon_i$$

This firefly algorithm has been implemented by many of the scholars for solving optimization problems, most of them have been changed into mathematical equations.

In this paper, the parameters of firefly algorithm are number of fireflies (m), number of generations / iterations (G), the light absorption coefficient (ϒ), randomization parameter (α), and attractiveness value ($\beta_{0)}$ have been defined.

- **Cuckoo Search Algorithm**

CA is one of the latest nature inspired algorithms, developed in 2009 by Yang and Deb (2009, 2010).It is based on the brood parasitism of some cuckoo birds by laying their eggs in the nests of other host birds and fruit flies [12].cuckoos are interesting birds, not only of their sounds they can make, but also of their reproduction. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to employ the new and potentially good solutions to replace the not-so-good solutions in the nests (12).

For simplicity in describing the standard cuckoo search, we introduce the following rules (13)

- Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
- High –quality eggs from the best nests will be carried over to the next generations
- The number of host nests available is fixed, and the laid egg by a cuckoo is found by the host species with a probality. In this case, the host species can either take the egg or simply abandon the nest and create a new nest.
- The probability of the discovery of an alien egg in its nest by a host bird is taken as $p_a \in [0, 1]$.

As a further approximation, this last assumption can be approximated by a fraction pa of the m host nests are replaced by new nests with new random solutions.

The pseudo code for cuckoo search algorithm 1 is as follows (7)

- Define an initialize parameters f(S), S=(S_1, S_2,….S_r)^T
- Generate initial population of m host nests S_i (i=1,2….m)
- While (t<maximum Generation) or (stop criterion) do
- Get a cuckoo randomly by Levy fights
- Evaluate its quality fitness Q_i
- Choose a nest among m (say, j) randomly
- If ($Q_i > Q_j$) then
- Replace j by the new solution
- End if
- Fraction (pa) of worse nests are abandoned and new ones built
- Keep the best solution (or nests with quality solutions)
- Rank the solutions and find the current best
- End while
- Post process results and visualization.

A new $S_i^{(p+1)}$ for cuckoo i is generated using a levy flight according to the following equation

$$S_i^{(p+1)} = S_i^{(p)} + \beta \wedge \text{Le'vy}(\lambda) \tag{7}$$

Where β (β> 0) is the final size that has to related to the problem of interest scale, The random walk described in Equation (7) is a Markov chain 's next location is dependent step length is drawn from a Levy distribution. The formula that describes the levy flight behavior in which the step lengths fit a probability distribution is

$$\text{Le'vy} \sim v = p^{-\lambda} \tag{8}$$

According to this formula, cuckoo bird's consecutive jumps or steps mainly form a random walking process that corresponds to a power –law –steps –length distribution with a heavy fail(2).

E. A Hybrid FA & CA algorithm for FOSSP

The following algorithm combines the advantages of cuckoo algorithm and firefly algorithm and to rectify the main disadvantage of CA easily becoming trapped in the local minima through the FA, which provides the local search faster than CA. The CA has only a single parameter, along with population size. So a hybrid algorithm is adopted to find better results. The main steps are introduced below.

Step 1: Initialization

Step 2: Generate the number of task T that need to br scheduled

Step 3: Schedule the task T using a Hybrid FA & CA as following

- At the beginning, the tasks are scheduled using cuckoo algorithm and the output result is produced using this algorithm.
- Then the output which is produced using CA is taken as an input for the FA and tasks are scheduled using FA.
- Finally, FA is again applied to the results that are produced in step 2.And final output result is given.

Step 4: Note down the processing time, resource utilization.

Step 5: Stop if the algorithm terminates. Else go to step 2 for scheduling the tasks. In the former, step3 is the main process of the algorithm.

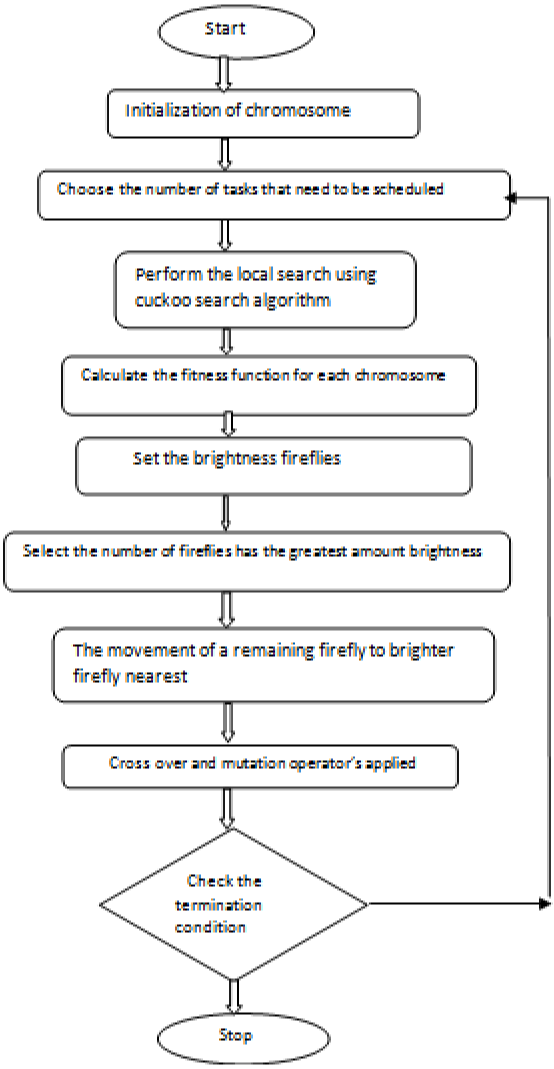

Figure 1: The Flow Chart of the Hybrid Algorithm (FA+CA) Method

Algorithm 2: Solving FOSSP Using Hybrid Algorithm

Initialization

Define the objective function f(S)

Generate initial population S_i, i=1,2,…m.

Light intensity /fitness value of population i is determined by objective function $f(S_i)$

Define the firefly algorithm parameters α,β_0,Υ

Define cuckoo algorithm

Levy fights with parameters β,λ.

Evaluate the makespan (fitness) of solutions Q_i

While (t < Max generation)

Get a cuckoo algorithm by Levy fights:

Evaluate its fitness or quality Q_i;

Choose a nest among m (say,j)randomly;

If ($Q_i > Q_j$) then

Replace j by the new solution

End if

For i= 1:m

For j= 1:m

Light intensity L(S) is determined by objective function f (S_i)

If $L_i < L_j$

Then move firefly i towards firefly j

(move towards brighter one)

End if

Attractiveness varies with distance d via $e^{-\Upsilon d}$, Evaluate new solutions and update light intensity.

End for j loop

End for i loop

Fitness assignment: Evaluate new solutions and update light intensity.

Stopping criterion: If the maximum number of generations has reached then terminate the search otherwise go to next iteration.

End while

IV. COMPUTATIONAL RESULTS

To prove the importance of the hybrid algorithm, it has implemented to different data set then used CA and FA for comparison in terms of make span and time needed to run the algorithms, the table 2 show the data set that contain 3 jobs and 3 machines.

Table1: An Example of 3 –Jobs 3-Machines Scheduling Problems with Processing Times [4]

Job	Operation	Time	Machine(M_k)		
J1 hn	O_{11}	6	6	-	-
J2					
J3					

The following table shows the time and makes span of hybrid algorithm with other well known algorithms of GA, ACO, Hybrid of GA &CA (3)

Table 2: Time and Make Span of the Algorithms

Algorithms	Time	Make Span
GA	0.133070	28
ACO	0.138325	28
CA+GA	0.056522	26
Hybrid algorithm(FA+CA)	0.032415	22

In addition we have taken different number of jobs and machines where number of jobs is greater than number of machines. The hybrid algorithm (FA+CA) gives better results than other well known results. (4)

Table 3: Time Taken by different Algorithms

No of Jobs	No of Machines	Time Taken by GA	Time taken by Hybrid (CA+GA)	Time Taken by Hybrid (FA+CA)
6	3	5.3191	0.3954	0.1274
16	6	1.4609	0.5137	0.4126
17	5	1.7534	0.5586	0.3672

The parameters for above table are taken as;

Maximum no of generation =5

No of domain =20

Number of machines = [3, 6, 5]

Number of jobs = [6, 16, 17]

The hybrid algorithms are implemented on very famous well known algorithms. In this section, some problems that were contributed to the OR – library are selected. The results are compared with other algorithm. The following table shows that the comparison of make span between the proposed algorithm and other algorithm designed by fisher and Thompson(TS),Lawrence (LW), Ala'a Abu-Srhahn and Muhannad AI –Hasan(AM).

Table 4: Comparison of Make Span between the Hybrid Algorithm and Other Algorithms

	Size	GA	ACO	GA+CA	Hybrid (3)Algorithm (CA+FA)
LW01	10x5	674	680	666	664
LW02	10x5	604	604	597	594
LW03	10x5	597	601	590	587
LW04	10x5	597	597	593	593
LW05	15x5	926	931	926	924
LW06	15x5	890	890	890	889
LW07	15x5	874	880	863	856
LW08	15x5	960	962	951	950

Table 4: Contd.,					
LW09	15x5	958	958	958	958
LW10	20x5	1154	1159	1150	1150
LW11	20x5	1300	1307	1292	1290
LW12	20x5	1209	1212	1207	1204
LW13	10x10	950	950	950	950
LW14	30x10	1903	1958	1888	1885
LW15	15x15	1279	1291	1268	1268
TS01	6X6	55	60	55	53
TS02	10X10	938	941	938	936
TS03	20X5	1169	1175	1165	1160
MT01	10X10	944	952	930	923
MT02	20X5	1169	1196	1165	1148

V. CONCLUSIONS

For the flexible open shop scheduling problem is one of the most important factors that the algorithm try to minimize it. A hybrid algorithm used here to minimize the make span for FOSSP. The disadvantage of cuckoo search algorithm is easily trapped by using firefly algorithm. The algorithms were tested using well known algorithms such as genetic algorithm, ant colony algorithm and with hybrid of CA & GA. The results shows that the hybrid with firefly and cuckoo algorithm yields the best solutions as measured by make span. The computational results show that the hybrid algorithm is more effective and gives better than the compared algorithms. All these challenging issues may motivate more research in the near future.

REFERENCES

1. *X.S.Yang,” Firefly algorithms for multi model optimization, stochastic Algorithms; Foundations and Applications”, Proceedings of the 5 th international conference Stochastic algorithms: foundations and applications,Sapporon,2009,pp.169-178.*
2. *M.Chandirasekaran,P.Ashokan, S. Kumanan, S. Arunnachalam, ” Application of selective breeding Algorithm for solving job shop scheduling problems”, International Journals of Manufacturing Science and Technology USA Vol 9(12), 2007,pp. 103-118.*
3. *M. Ravichandran, M.Omkumar,”Hybrid flow shop scheduling using improved hybrid ACO cuckoo algorithm to minimize make span”, International journal of computer Applications”, Vol.115 (18), 2015, pp. 16-20.*
4. *Ala’a Abu-Srhahn and Muhannad AI –Hasan,” hybrid algorithm using Genetic algorithm and cuckoo search algorithm for job shop scheduling problem”, International journal of computer science, Vol. 12(2),2015, pp.1694-0784.*
5. *R.G.Babukartik,P.Dhavachelvan”,hybrid algorithm using the advantages of ACO and cuckoo search for job scheduling”, International Journal of information technology convergence and services, Vol.2(5),2012 pp.51- 60*
6. *Satyendra Singh, Jitendra Kurmi,Sudanshu Prakash Tiwari”, A hybrid genetic and cuckoo search Algorithm for job scheduling”, International journal of scientific and research publications, Vol. 5(6), 2015, pp. 1-4.*
7. *Sharik Farook, p.sangameswara Raju, ”Evolutionary hybrid genetic –firefly algorithm for global optimization”, International journal of computational engineering &management, Vol 16(3),2013.pp. 37-45.*
8. *B.mohammed Tharik,” A comparative study of firefly algorithm and cuckoo search algorithm in Optimizing turning operation with constrained parameters”, International journal of engineering research & technology Vol 2(4),2013.pp.1701- 1706.*

9. *Lina Zhang, Liqiang Liu, Xin –she Yang, Yuntao Dai, "A novel hybrid firefly algorithm for global optimization", Plos one,2016,pp.1-17.*

10. *Maheshwar,Sapna Bhatt, Ramesh Kumar, "Firefly algorithm based improved genetic algorithm :a proposed approach", International journal of engineering research &technology. Vol 3(7), 2014.pp.1537-1539.*

11. *Hossein Parven, Ebrahim Behrouzian Nejad, Seyed Enayatolah Alavi, "A new hybrid algorithms for task scheduling in computational grids to decrease makespan", International journal of computer science & network solutions, Vol 2 (4),pp. 19-28.*

12. *Sumandeep Aujila, Amandeep Ummat, " Task scheduling in cloud using hybrid cuckoo algorithm", International journal of computer networks and applications", Vol.2(3), 2015,pp.144-150.*

13. *Xin-she yang, suash Deb, "Cuckoo search :Recent advances and applications", Neural Computing and Applications, Vol 24,2014,pp.169-174.*

14. *Hui Nie and Bo Liu, Purmo Xie, Zhenbing Liu and Huihua Yang, "A cuckoo search algorithm for scheduling multiskilled workforce", Journal of networks,Vol 9(5),2014, pp.1346-1353.*

15. *Aphirak Khadwilard, Sirikarn Chansombat, Thatchai Thepphakorn, "Application of firefly Algorithm and Its parameter setting for job shop scheduling", The journal of Industrial Technology Vol 8(1),2012,pp.2555-2564.*

16. *Alireza khatami, Seyed Habib A. Rahmati "An efficient firefly algorithm for the flexible job shop scheduling problem", International conference on Industrial Engineering and Operations Management Dubai,United Arab Emirates (UAE) Vol 5, 2015,pp. 2144-2146.*

17. *S. Karthikeyan,P. Asokan, S. Nickloas, "A hybrid discrete firefly algorithm for multi objective flexible job shop scheduling problem with limited resource constraints", International Journal of Advanced Manuf Technol", Vol 72,2014,pp.1567 – 1579.*

18. *X-S. Yang, "Nature inspired metaheuristic Algorithms, Luniver Press", Vol 1, 2008, pp. 60-78.*

19. *K.C. Udaiyakumar, M. Chandrasekaran," Application of firefly algorithm in job shop scheduling problem for minimization of makespan" Procedia Engineering - Elesvier Vol 97,2014,pp.1798-1807.*

20. *J.Gao,L.Sun,and M. Gen," A hybrid genetic and variable neighbourhood descent algorithm for flexible job shop scheduling problem," Computers operation Research, Vol 35(9),2008,pp.2892-2907.*

21. *Iztok Fister, Iztok Fister Jr, Xin –She Yang ,Janez Bress, "A comprehensive review of firefly algorithm.","Swarm &Evolutional computation", Elsevier, Vol 13,2013, pp. 34-46.*

22. *Dinakara Prasad Reddy P, J N Chandra Sekar "Application of firefly Algorithm for combined Economic Load and Emission Dispatch " International journal on Recent and Innovation Trends in Computing and Communication,Vol 2(8),2014, pp. 2448-2452.*

23. *M.K. Sayadi, R. Ramezanin, N. Ghaffari-Nassab, "A discrete firefly meta-heuristic with local search for make span minimization in permutation flow shop scheduling problems", International Journals of Industrial Engineering Computations, Vol 1(1), 2010, pp. 1-10.*